
Elektrotehnički fakultet u Beogradu
Katedra za računarsku tehniku i informatiku

Predmet: Objektno orijentisano programiranje
(OE2OOP, OE4OOP, OF2OO1, OS2OO1, OS3OOP,
OT2OOP, OT3OOP)
Nastavnik: Prof. dr Dragan Milićev
Ispitni rok: Oktobar 2010.
Datum: 21.09.2010.

Kandidat: _____

Broj Indeksa: _____ *E-mail:* _____

*Ispit ima 2 dela ukupnog trajanja 3 sata. Na prvom delu **nije** dozvoljeno korišćenje literature. Na drugom delu **jeste** dozvoljeno korišćenje literature. Trajanje prvog dela student određuje prema ličnom nahođenju. Student može početi da koristi literaturu kad preda prvi deo.*

UPISATI SVOJE PODATKE I NA PRVOJ STRANI DRUGOG DELA ISPITA!

Pismeni ispit:

Zadatak 1 _____/10
Zadatak 2 _____/10
Zadatak 3 _____/10
Zadatak 4 _____/10
Zadatak 5 _____/10

Rad u toku semestra:

Projekat _____/20
Domaći zadaci _____/30

Ukupno na ispitu: _____/50

Ukupno u toku semestra: _____/50

Ukupno: _____/100

Ocena: _____ (_____)

Napomena: Ukoliko u zadatku nešto nije dovoljno precizno definisano, student treba da uvede razumnu pretpostavku, da je uokviri (da bi se lakše prepoznala prilikom ocenjivanja) i da nastavi da izgrađuje preostali deo svog odgovora na temeljima uvedene pretpostavke. Ocenjivanje unutar potpitanja je po sistemu "sve ili ništa", odnosno nema parcijalnih poena. Kod pitanja koja imaju ponuđene odgovore treba **samo zaokružiti** jedan odgovor. Na ostala pitanja odgovarati **čitko, kratko i precizno.**

1. (10 poena)

Dati su sledeći delovi koda na jeziku C++ (svaki deo je u posebnom fajlu):

(a)

```
enum Color {red, green, blue};  
Color yellow = green + blue;
```

(b)

```
enum Color {red, green, blue};  
int yellow = green + blue;
```

(c)

```
class B {  
public: B(int=0);  
};
```

```
class D : public B {  
public: D (int) {}  
};
```

(d)

```
class X;  
X& p = *new X;
```

(e)

```
class X {};  
X* p = new X;
```

Za svaki od ovih delova koda u donju tabelu upisati da li će prevodilac prijaviti grešku ili neće i dati kratko obrazloženje o uzroku greške ili kratko obrazloženje zašto greške nema:

Deo koda	Ima ili nema greške? Napisati kratko obrazloženje
a	
b	
c	
d	
e	

2. (10 poena)

Sledeći program se ispravno prevodi, ali ima jednu logičku grešku pri izvršavanju.

```
#include <iostream>
using namespace std;

class Widget {
public:
    static Widget* create(int);
    virtual ~Widget() { cout << "Widget\n"; }
    Widget() {}
};

class Gadget : public Widget {
protected:
    ~Gadget () { cout << "Gadget\n"; }
};

class Doodad : public Widget {
protected:
    ~Doodad () { cout << "Doodad\n"; }
};

Widget* Widget::create (int i) {
    if (i%2) return new Gadget();
    else return new Doodad();
}

int main () {
    const size_t WIDGET_COUNT = 5;
    Widget *magicBox[WIDGET_COUNT];
    for (int i=0; i<WIDGET_COUNT; i++)
        magicBox[i] = Widget::create(i);
    return 0;
}
```

- (a) Šta ispisuje dati program?
- (b) Prokomentarišite ovaj program u smislu iznad pomenute greške. Šta treba promeniti u ovom programu i/ili dodati u ovaj program da bi greška bila uklonjena?
- (c) Po uklanjanju greške, šta ispisuje popravljani program?

3. (10 poena)

Na jeziku C++ realizovati klasu `Point` koja apstrahuje tačku u trodimenzionalnom Dekartovom koordinatnom sistemu. Tačka se predstavlja koordinatama (x,y,z) tipa `double`. Ova klasa treba da obezbedi sledeće:

- Inicijalizaciju koordinatama, pri čemu je podrazumevana inicijalizacija na $(0,0,0)$.
- Operacije poređenja na jednakost (`operator==`) i nejednakost (`operator!=`) dve tačke.
- Operaciju izračunavanja rastojanja (`operator-`) dve tačke.
- Operaciju koja vraća tačku simetričnu datoj tački u odnosu na koordinatni početak (`operator-`).
- Operaciju pomeranja tačke za $(\Delta x, \Delta y, \Delta z)$.

4. (10 poena)

Projektuje se sistem za proračunavanje karakteristika zračenja radio antena određenog oblika. *Antena (Aerial)* ima, kao svojstva, visinu i poluprečnik (racionalni brojevi u odgovarajućim jedinicama mere), kao i proračunatu karakteristiku zračenja (niz racionalnih brojeva veličine `MAX_SIZE`). Antena ima i operaciju `computeCharacteristics()` koja proračunava karakteristiku date antene sa datim svojstvima i rezultat proračuna smešta u navedeni niz. Proračun karakteristike se može vršiti po više metoda, označenih sa *MethodA*, *MethodB* i *MethodC*. Potrebno je obezbediti mogućnost da se različiti primerci antena mogu konfigurisati tako da se za njih izračunavaju karakteristike različitim metodama, s tim da je skup metoda lako proširiv bez izmene ostatka sistema, a da je konfigurisanje metoda za date primerke antena izvodivo i dinamički (u vreme izvršavanja).

Ako je korišćen neki projektni obrazac, navesti koji i koje klase u ovom sistemu igraju koje apstraktne uloge definisane tim projektnim obrascem.

- (a) Nacrtati dijagrame koji prikazuju UML model klasa opisanog sistema, sa navođenjem naziva metoda pomenutih u opisu. Objasniti koji je DP uočljiv u postavci zadatka i objasniti koja klasa koju ulogu igra u uočenom DP.
- (b) U potpunosti implementirati klasu `Aerial` na jeziku C++. Dati primer nekoliko objekata klase `Aerial` konfigurisanih sa različitim metodama proračuna.

5. (10 poena)

U nekom školskom programu, koji služi za objašnjavanje OO koncepata kroz rad sa zbirkama geometrijskih figura u ravni, podržano je nekoliko tipova figura (krug, kvadrat, trougao). Figure su određene koordinatama težišta i odgovarajućim dimenzijama (krug – poluprečnik, kvadrat – dužina stranice, trougao – dužina sve tri stranice). Program je po vrsti konzolna aplikacija interaktivnog tipa. Program može raditi sa više zbirki istovremeno. Korisnik može birati tekuću zbirku, dodavati figure u tekuću zbirku figura i uklanjati ih iz zbirke, kao i ispisivati podatke o figurama trenutno sadržanim u tekućoj zbirci figura. Korisnik može po svojoj želji ispisivati različite osobine trenutno dostupnih figura i zbirki figura. Korisnik u meniju bira ispis željene osobine, nakon čega program obilazi sve objekte napravljene do tog trenutka. Potrebno je napisati samo deo programa koji služi da korisniku omogući proizvoljne obilaske svih napravljenih objekata, kao i podršku za funkcionalnosti koje ilustruju taj obilazak. Konkretno funkcionalnosti su: ispis površine svih postojećih objekata i prebrojavanje objekata sa obimom većim od zadatog. Kod ispisa površine, za svaki objekat ispisati vrstu i površinu (površinu zbirke određivati kao zbir površina sadržanih objekata). Kod prebrojavanja, zbirke treba da ispišu broj sadržanih figura koje zadovoljavaju kriterijum, a figure ne treba da ispišu ništa. Obim zbirke ne treba razmatrati. Na kraju prebrojavanja, program treba da ispiše ukupan broj figura. Pretpostaviti da je ostatak programa opisanog u postavci napisan i da su klase kojima su figure predstavljene opremljene metodama za određivanje površine, obima i udaljenosti težišta od koordinatnog početka. Takođe, pretpostaviti da su sve klase opremljene ispravno napisanim konstruktorima kopije, operatorima dodele i destruktorima, te da je ostatak programa napisan tako da ispravno koristi dinamičku memoriju.

- (a) Nacrtati kompletan UML dijagram klasa povezanih sa obilaskom objekata.
- (b) Napisati kompletan C++ kod metode koja će biti pozvana prilikom ispisa površine zbirke figura.
- (c) Napisati kompletan C++ kod koji pokreće opisano prebrojavanje i vrši ispis ukupnog broja figura sa obimom većim od zadatog.
- (d) Koji je projektni obrazac ovde primenjen?